

up: [Chapter 5 -- Memory Management](#)

prev: [5.1 Segment Translation](#)

next: [5.3 Combining Segment and Page Translation](#)

---

## 5.2 Page Translation

In the second phase of address transformation, the 80386 transforms a linear address into a physical address. This phase of address transformation implements the basic features needed for page-oriented virtual-memory systems and page-level protection.

The page-translation step is optional. Page translation is in effect only when the PG bit of CR0 is set. This bit is typically set by the operating system during software initialization. The PG bit must be set if the operating system is to implement multiple virtual 8086 tasks, page-oriented protection, or page-oriented virtual memory.

### 5.2.1 Page Frame

A page frame is a 4K-byte unit of contiguous addresses of physical memory. Pages begin on byte boundaries and are fixed in size.

### 5.2.2 Linear Address

A linear address refers indirectly to a physical address by specifying a page table, a page within that table, and an offset within that page. [Figure 5-8](#) shows the format of a linear address.

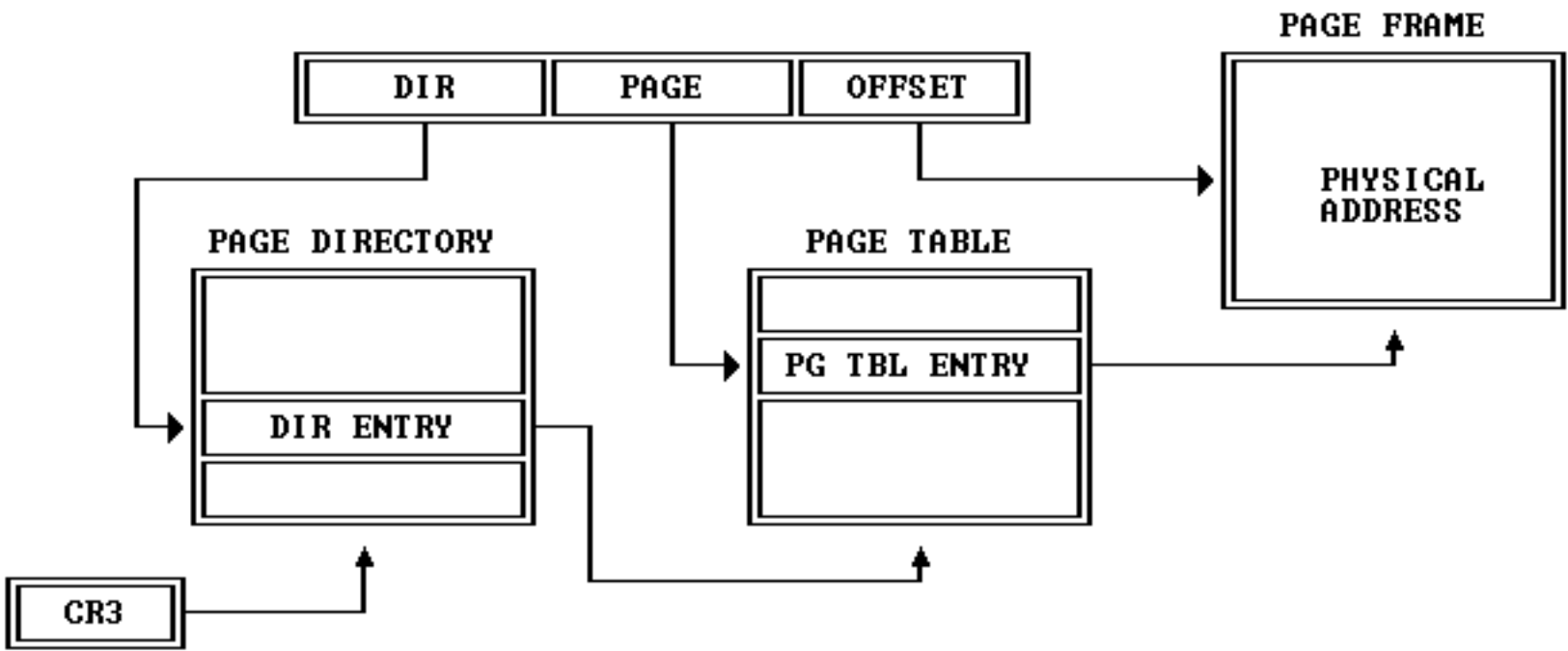
[Figure 5-9](#) shows how the processor converts the DIR, PAGE, and OFFSET fields of a linear address into the physical address by consulting two levels of page tables. The addressing mechanism uses the DIR field as an index

into a page directory, uses the PAGE field as an index into the page table determined by the page directory, and uses the OFFSET field to address a byte within the page determined by the page table.

Figure 5-8. Format of a Linear Address



Figure 5-9. Page Translation



### 5.2.3 Page Tables

A page table is simply an array of 32-bit page specifiers. A page table is itself a page, and therefore contains 4 Kilobytes of memory or at most 1K 32-bit entries.

Two levels of tables are used to address a page of memory. At the higher level is a page directory. The page directory addresses up to 1K page tables of the second level. A page table of the second level addresses up to 1K pages. All the tables addressed by one page directory, therefore, can address 1M pages ( $2^{20}$ ). Because each page contains 4K bytes ( $2^{12}$  bytes), the tables of one page directory can span the entire physical address space of the 80386 ( $2^{20}$  times  $2^{12} = 2^{32}$ ).

The physical address of the current page directory is stored in the CPU register CR3, also called the page directory base register (PDBR). Memory management software has the option of using one page directory for all tasks, one page directory for each task, or some combination of the two. Refer to [Chapter 10](#) for information on initialization of CR3. Refer to [Chapter 7](#) to see how CR3 can change for each task.

## 5.2.4 Page-Table Entries

Entries in either level of page tables have the same format. [Figure 5-10](#) illustrates this format.

### 5.2.4.1 Page Frame Address

The page frame address specifies the physical starting address of a page. Because pages are located on 4K boundaries, the low-order 12 bits are always zero. In a page directory, the page frame address is the address of a page table. In a second-level page table, the page frame address is the address of the page frame that contains the desired memory operand.

### 5.2.4.2 Present Bit

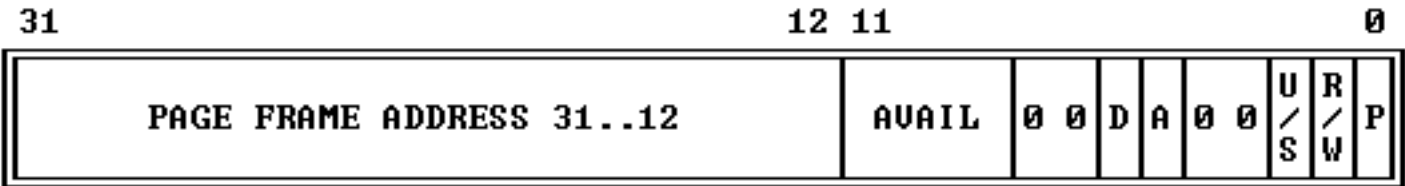
The Present bit indicates whether a page table entry can be used in address translation. P=1 indicates that the entry can be used.

When P=0 in either level of page tables, the entry is not valid for address translation, and the rest of the entry is available for software use; none of the other bits in the entry is tested by the hardware. [Figure 5-11](#) illustrates the format of a page-table entry when P=0.

If P=0 in either level of page tables when an attempt is made to use a page-table entry for address translation, the processor signals a page exception. In software systems that support paged virtual memory, the page-not-present exception handler can bring the required page into physical memory. The instruction that caused the exception can then be reexecuted. Refer to [Chapter 9](#) for more information on exception handlers .

Note that there is no present bit for the page directory itself. The page directory may be not-present while the associated task is suspended, but the operating system must ensure that the page directory indicated by the CR3 image in the TSS is present in physical memory before the task is dispatched . Refer to [Chapter 7](#) for an explanation of the TSS and task dispatching.

Figure 5-10. Format of a Page Table Entry



- P - PRESENT
  - R/W - READ/WRITE
  - U/S - USER/SUPERVISOR
  - D - DIRTY
  - AVAIL - AVAILABLE FOR SYSTEMS PROGRAMMER USE
- NOTE: 0 INDICATES INTEL RESERVED. DO NOT DEFINE.

Figure 5-11. Invalid Page Table Entry



### 5.2.4.3 Accessed and Dirty Bits

These bits provide data about page usage in both levels of the page tables. With the exception of the **dirty bit** in a **page directory entry**, these bits are **set by the hardware**; however, the **processor does not clear** any of these bits.

The processor sets the corresponding accessed bits in both levels of page tables to one before a read or write operation to a page.

The processor sets the dirty bit in the second-level page table to one before a write to an address covered by that page table entry. The dirty bit in directory entries is undefined.

An operating system that supports paged virtual memory can use these bits to determine what pages to eliminate from physical memory when the demand for memory exceeds the physical memory available. The operating system is responsible for testing and clearing these bits.

Refer to [Chapter 11](#) for how the 80386 coordinates updates to the accessed and dirty bits in multiprocessor systems.

### 5.2.4.4 Read/Write and User/Supervisor Bits

These bits are not used for address translation, but are used for **page-level protection**, which the processor performs at the same time as address translation . Refer to [Chapter 6](#) where protection is discussed in detail.

## 5.2.5 Page Translation Cache

For greatest efficiency in address translation, the processor stores the most recently used page-table data in an on-chip cache. Only if the necessary paging information is not in the cache must both levels of page tables be referenced.

The existence of the page-translation cache is invisible to applications programmers but not to systems programmers; operating-system programmers must flush the cache whenever the page tables are changed. The page-translation cache can be flushed by either of two methods:

1. By reloading CR3 with a [MOV](#) instruction; for example:

```
MOV CR3, EAX
```

2. By performing a task switch to a TSS that has a different CR3 image than the current TSS . (Refer to [Chapter 7](#) for more information on task switching.)

---

**up:** [Chapter 5 -- Memory Management](#)

**prev:** [5.1 Segment Translation](#)

**next:** [5.3 Combining Segment and Page Translation](#)